Lane Detection (Part I): Mono-Vision Based Method

Hao LI, Fawzi NASHASHIBI

TECHNICAL REPORT

N° 433

January 2013

Project-Team IMARA

ISSN 0249-6399 ISRN INRIA/RT--433--FR+ENG

Lane Detection (Part I): Mono-Vision Based Method

Hao LI[*], Fawzi NASHASHIBI[†]

Project-Team IMARA

Technical Report n° 433 — January 2013 — 29 pages

Abstract: In this report, we introduce a mono-vision based lane detection method (pure mono-vision without the aide of any other sensor). The lane detection method consists of the sub-parts of lane mark segment extraction, lane model fitting, and lane tracking. The lane mark segment extractor is based on some criteria that are only related to the inherent physical properties of the lane marks. The lane model is parameterized in the world (the ground plane) reference instead of in the image reference. The lane tracking is carried out in the framework of the particle filter, including the evolution and the update of the lane state; a mechanism that executes on-line estimation of the vehicle pitch angle is naturally incorporated in the lane tracking process. Read-data tests are given to demonstrate the performance of the mono-vision based lane detection method.

Key-Words: lane detection, mono-vision, inverse perspective mapping, lane tracking, particle filter

[*] Hao LI: Imara, INRIA Paris-Rocquencourt, hao.li@inria.fr
[†] Fawzi NASHASHIBI: Imara, INRIA Paris-Rocquencourt, fawzi.nashashibi@inria.fr

RESEARCH CENTRE
ROCQUENCOURT

Domaine de Voluceau
Rocquencourt BP 105
78153 Le Chesnay Cedex France

Reconnaissance de Voie (Part I) : Méthode Fondée Sur Vision-Monoculaire

Résumé : Dans ce rapport, on a introduit une méthode de la reconnaissance de voie fondée sur vision-monoculaire (pure vision-monoculaire sans l'aide de aucune capteur). La méthode de la reconnaissance de voie consiste des sous-parts de l'extraction de marquages routiers, ajustement du modèle de voie, et le pistage de voie. L'extracteur de marquages routiers se fonde sur des critères qui ne sont reliés qu'aux traits physiques inhérents des marquages routiers. Le modèle de voie se paramètre dans le repère mondial (le plan du sol) au lieu du repère de l'image. Le pistage de voie se met en œuvre dans l'architecture du filtre particulaire, y compris l'évolution et la mise à jour de l'état de voie ; un mécanisme qui exécute l'estimation en ligne de l'angle tangage du véhicule est intégré naturellement dans le processus du pistage de voie. L'expérimentation aux données réelles a été conduite pour montrer la performance de la méthode de la reconnaissance de voie fondée sur vision-monoculaire.

Mots-Clés : reconnaissance de voie, vision-monoculaire, projection perspective inverse, pistage de voie, filtre particulaire

1 Introduction

Lane detection, as the name itself indicates, is a process of detecting (as well as recognizing) the lanes where the ground traffic circulates. For humans, lane detection is one of the essential functions related to the driving experience. As artificial intelligence technologies find its way into the transportation field, lane detection becomes more and more a specific term that implies the utilization of certain perceptive sensors, certain processing units, and certain algorithms to perform this functionality automatically; in simple words, we let a sensor-computer system itself to perform lane detection.

For human vision and human intelligence, the task of lane detection is usually a piece of cake; when facing the variations and even severe variations of road conditions, such as the appearance of tree shadows, the existence of surrounding objects, the change of light condition, the dirt left on the road surface and etc, we humans still have no difficulty in detecting the lane marks. In contrast, all these variations of road conditions pose challenging factors for automatic lane detection based on a sensor-computer system (latter, we still use the short terms "lane detection" without always mentioning the sensors and the computers).

Despite the existence of an ocean of research works on lane detection [1], some of which can even be dated back to many decades ago, the difficulties in lane detection always exist. By so far, no existing technique can boast of detecting the lanes successfully in all circumstances where the lanes are detectable by us humans.

We might think that the difficulties in lane detection only reflect the immaturity of current techniques. Remember that before the invention of the vehicle, being able to move faster than a horse was also regarded as difficult, but no longer is. Naturally, we might simply expect the appearance of a perfect technique that will sweep away all nowadays difficulties in lane detection. However, before having such an expectation, we had better understand the causes for these difficulties.

One cause, as mentioned above, is the large variability of road conditions. Imaginarily given an ideal situation where the road background is always monotonous without any road side object, where the lane marks are always the only bright white features, and where the lighting condition is always smooth, lane detection would be rather an easy task.

Another cause for the difficulties in lane detection is due to the ambiguity of the concept "lane" itself from the perspective of ontology. Unlike those entities whose definition does not depend on the environment, the "lane" forms itself together with the environment. For example, a human being, no matter walking on a street, climbing on a cliff, or sleeping on a bed, is always regarded as a human being. Whereas for a lane that

consists of painted lane marks, if we paint the same lane marks (i.e. a group of white strips) on a wall, they would appear more like a kind of decoration instead of arousing the concept "lane" in our mind. Therefore, a lane is in some sense *ad hoc* defined by its environment which can be of immense sorts. In other words, the definition of the lane is rather data-driven (the data of the whole environment including the lane) than model-driven (an underlying mechanism that forms an entity).

The data-driven characteristic of the lane definition can also be reflected by one of our daily experiences. Sometimes, we will pass a street with road construction works, where new lane marks are painted to rearrange the traffic temporarily but old lane marks might still be left there. I believe that no one will make mistakes in following the correct lane in such a scenario. However, what is the essential difference between the two sets of lane marks themselves?

Because of the data-driven characteristic of the lane definition, the ambition of finding an omnipotent lane detection technique is equivalent to being able to figure out a technique that can correctly interpret all kinds of the environment containing the lane. This requires not only an enhancement of computational resources but also a biological breakthrough on understanding how we understand the environment, which obviously has not been fulfilled by so far.

Therefore, we currently have no ambition to figure out a lane detection technique that is perfectly intelligent, i.e. being able to detect the lane correctly (neither false positive nor true negative) for all kinds of data. What we intend to achieve is to design a technique that tries to fit reasonable lane boundaries to lane mark-like features. We use "reasonable" to indicate that the detection result should be consistent with common a priori knowledge of a lane configuration. The reason to use "lane mark-like features" here instead of directly using "lane marks" is to indicate that we have no intention to interpret the same entity appearances differently according to their different environments. In the following section, we will present our methodology of lane detection together with the analysis based on state-of-the-arts.

2 Mono-Vision Based Method

If we focus on detecting the lanes characterized by visual lane marks, then vision (cameras) will be indispensable in lane detection. Vision based sensing is most related to our own sensing experience, compared with range data (laser rangefinder) based sensing, magnet based sensing and etc.

As reported in literature, researchers always try to base their lane detection components only on vision (perhaps together with motion sensors) and out of "economical" consideration try to exclude other sensors such as laser rangefinder and GPS. On the

other hand, we had better bear in mind that different vehicular intelligent functions are not isolated from each other and that a sensor has the potential to serve for not only one single function. As explained in [2], it has been a tendency to incorporate different types of sensors (vision, laser scanner, motion sensor, and GPS) into a vehicle, in order that the vehicle possesses fairly complete intelligent perception ability towards itself and the environment. It would be valuable to consider a function from the perspective of the integration of the entire vehicle system, not only from the function itself. Concerning lane detection, not only vision but also other sensors can be utilized.

Multi-sensor fusion based method has apparent advantage over pure vision-based method. For example, with range data, the image area associated with non-lane mark objects can be easily excluded, which can largely reduce misleading factors in the vision based lane detection. However, we would like to postpone the presentation of a multi-sensor fusion based methodology to our next report ("*Lane Detection (Part II): Multi-Sensor Fusion Based Method*"). In this report, we exploit the possibility of performing lane detection based on minimum sensor requirements. Since a lane detection system requires at least one mono-vision, we focus on only using mono-vision to perform lane detection, even without the aide of vehicle motion sensors—In contrast, the availability of vehicle motion data is usually taken for granted in many existing works. In other words, we try to maximize the capability of the pure mono-vision based method, which is not intended to serve as a final solution but serve naturally as a base stone for extending the works into a multi-sensor fusion based methodology.

2.1 Lane mark feature extraction

Few methods try to handle the image on the whole and output the lane estimate in a mechanism similar or at least intended to be similar to that of humans. A typical example is the neural networks system (*ALVINN: Autonomous Land Vehicle In a Neural Networks*) [3], which incorporates the image as an input on the whole and directly outputs vehicle control commands. Such kind of machine learning based systems that try to function in a human manner might be our dream in the long run. For the moment, the challenging issues such as the large amount of training data, how to select the training data, and how to guarantee the quality of the training data hinder the application of such kind of methods.

For overwhelming part of existing lane detection methods, the first fundamental process is the extraction of certain lane mark features. Some methods rely on explicit heuristic rules. For example, observing that a lane mark is usually a bright strip with comparatively darker local background, we might extract all those image points with strong intensity contrast against their two neighboring sides [4]. Similarly, observing

that a sharp intensity contrast usually exists at the edge of lane marks, we might extract strong edge points (above certain threshold) or might only extract the strongest edge point in certain local image patches [5]. Some methods rely on implicit rules that are tuned by machine learning methods, such as support vector machine based classifier [6]. As lane marks are rather line-segment like features than point like features, some methods extract line segment features instead of extracting point features. A popular technique for extracting the line segments is the Hough transform [7]. Local edge direction estimated according to the gradients can be used to reduce the searching space of the Hough transform [8].

A common limitation of existing methods is that the criteria used to select lane mark features are not related to the inherent physical properties of the lane marks. For example, light intensity contrast between lane mark points and their surrounding background is often employed as the criteria; however, it is not an inherent property of the lane marks and a lane mark extractor based on it is often susceptible to the variation of lighting conditions. Those criteria associated with machine learning based methods are even far away from the physical properties of the lane marks, not to mention the inherent physical properties. Because of this limitation, existing methods usually require tuning of certain parameters (either according to heuristic experiences or according to certain machine learning rules) that rather depend on concrete application scenarios.

We intend to design a lane mark feature extractor that is related to the inherent physical properties of the lane marks. More specifically, we use an edge detector with a low threshold to guarantee extracting the contours of the candidate lane mark segments; then we use general a priori knowledge about the inherent physical properties of the lane marks to sift away those unqualified lane mark segment candidates.

2.2 Lane model fitting

Another fundamental process after lane mark feature extraction is lane model fitting (or lane hypothesis). The requirement for lane model fitting is two-folds: first, lane model fitting can propose a continuous lane configuration out of discreet lane mark features; second, it can smooth the errors of the lane mark features.

Line model can be fairly used to fit lane marks in a short range in front of the vehicles. For certain applications such lane departure warning, the line model fitting results are acceptable for fulfilling the tasks. On the other hand, for lane detection up to a range say 40 meters in front of the vehicle, the limitation of a line model is obvious.

To increase the representation ability of the lane model, the parabolic curve model [9] and the spline model [10] have been used for fitting the lane mark features. A challenging issue for the lane model fitting process is the computational cost. For real-

time feasibility, it is unlikely to traverse the entire lane model parameter space, especially when the lane model is complicated. To reduce the computational cost, partial-searching methods such as the Metropolis algorithm [9], the random Hough transform [7], and the RANSAC technique [11] have been used to search for semi-optimal solutions (*RANSAC: RANdom SAmple Consensus* [12]).
In our method, we parameterize the lane model in the world (the ground plane) reference instead of in the image reference. We borrow the RANSAC spirit to generate lane configuration hypothesis.

2.3 Lane tracking

When we perceive a lane with our own vision, there is no apparent trace of visual "tracking" in this process: our ability to perceive the lane at any instant does not depend on our perception results at previous instants.
On the other hand, lane tracking is usually an important part in artificial vision based lane detection. It is intended to enhance the efficiency and the robustness of the lane detection process. The lane estimation in the last system period can be used to specify the regions of interest in current system period, which not only can reduce the searching space for the lane mark features but also can reduce the noise-signal ratio. Besides, lane detection can be regarded to establish a temporal consensus among the lane detection results at different times, which can reduce the false positives that appear coincidentally.
The convenient Kalman filter (or extended Kalman filter) that is popular in many fields also finds its application in the process of lane tracking [13]. On the other hand, the particle filter (sampling based method) is also a powerful tool for lane tracking [14].
Compared with the Kalman filter, the particle filter has two noticeable merits: first, it has the potential to model arbitrary distributions; second, as an extension of the first merit, it can naturally handle multi-modal states. For example, in ambiguous situation where it is difficult to determine the correct one out of two lane configuration hypotheses that are both plausible, the particle filter allows to temporarily keep both hypotheses for further judgements based on new upcoming data. In contrast, the Kalman filter requires a definitive choice between the two hypotheses, and might make the wrong choice, which will incur the kidnapping problem. We prefer to use the particle filter as the technique for lane tracking.

3 Lane Mark Segment Extraction

From this section, we will introduce in details our mono-vision based lane detection method.

3.1 Inverse perspective mapping (IPM)

Image processing is carried out on the bird-eye view image instead of directly on the camera image; the bird-eye view image is obtained from the camera image by inverse perspective mapping (details will be given in this sub-section latter), see Fig.1 (b). The practice of detecting lane mark features from the bird-eye view image can be found in literature, such as in [4] [6] [14]. Compared to the original camera image, the bird-eye view image possesses two merits for lane detection: first, any processing result on the bird-eye view image can be related to actual physical properties of 3D world entities; second, lane marks are almost aligned in the same direction in the bird-eye view image, which brings convenience for image processing.

Figure 1 (a: left) camera image; (b: right-top) bird-eye view image; (c: right-middle) edge segments; (d: right-bottom) lane mark segments

Let the vehicle be stationary on a flat ground, the vehicle coordinate system (or the vehicle reference) is established in a way that the X-axis is pointing longitudinally forward, the Z-axis is pointing vertically upward, and the Y-axis is determined by the right-hand rule; besides, the X-O-Y plane is situated on the ground plane. Such way of establishing the vehicle coordinate system is for the convenience of relating the image data to the ground plane. See Fig.2.

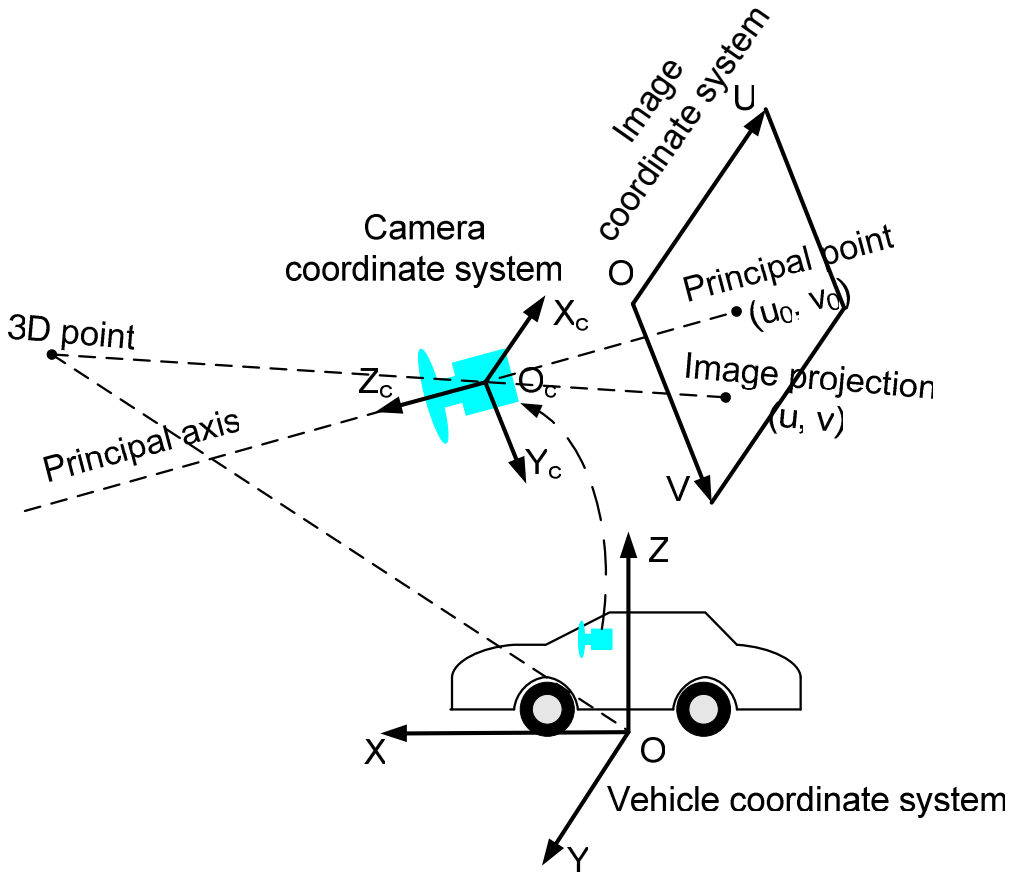

Figure 2 the image coordinate system and the vehicle coordinate system

The camera intrinsic parameters are always fixed and only need to be calibrated once using say the standard chessboard based method [15]. As the camera is fixed on the vehicle, the spatial relationship between the camera coordinate system and the vehicle coordinate system is also fixed and the camera extrinsic parameters can also be determined. In practice, it is not necessary to make a clear distinction between the camera intrinsic parameters and the camera extrinsic parameters; we can express the relationship between the vehicle coordinate system and the image coordinate system in a compact form of *perspective mapping* [16] (attention to the difference between the camera coordinate system and the image coordinate system which are related by the camera intrinsic parameters to each other):

$$\beta \mathbf{P}_I = \mathbf{M} \cdot \mathbf{P}_V$$

$$\beta\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} m_1 & m_2 & m_3 & m_4 \\ m_5 & m_6 & m_7 & m_8 \\ m_9 & m_{10} & m_{11} & m_{12} \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{3-1}$$

The matrix **M** is called the *perspective matrix*; $\mathbf{P}_I$ denotes the augmented image coordinate vector; $\mathbf{P}_V$ denotes the augmented vehicle coordinate vector. Due to the movement of the vehicle, the X-O-Y plane is not always ideally aligned with the ground plane; there might be slight discrepancy between the two planes. According to our observation, the pitch angle usually accounts for the considerable part of this discrepancy. We can use a simplified model to describe the influence of the pitch angle (denoted as α); as shown in Fig.3.

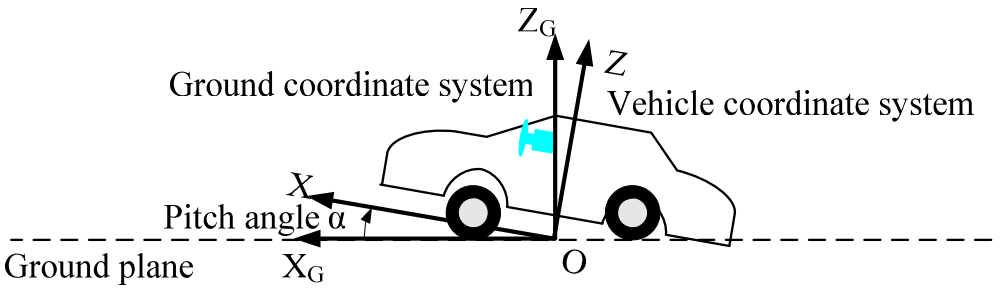

Figure 3 the discrepancy caused by the pitch angle

The relationship between the ground plane points and their image projects are given as follows:

$$\beta\mathbf{P}_I = \mathbf{M}\cdot\mathbf{P}_V = \mathbf{M}\cdot\mathbf{R}_y(-\alpha)\cdot\mathbf{P}_G = \mathbf{M}_\alpha\cdot\mathbf{P}_G \tag{3-2}$$

$$\mathbf{R}_y(-\alpha) = \begin{bmatrix} \cos(-\alpha) & 0 & -\sin(-\alpha) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(-\alpha) & 0 & \cos(-\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};\ \mathbf{M}_\alpha = \mathbf{M}\cdot\mathbf{R}_y(-\alpha);\ \mathbf{P}_G = \begin{bmatrix} x_g \\ y_g \\ z_g \\ 1 \end{bmatrix}$$

Given an arbitrary point $\mathbf{P}_G$ on the ground plane (z_g=0), we can compute its image counterpart $\mathbf{P}_I$ via (3-2). Vice versa, given an image point $\mathbf{P}_I$ and we suppose it situated on the ground plane, we can also compute its world counterpart $\mathbf{P}_G$—Attention: given a general image point, it is impossible to determine its world counterpart. Since we focus on lane detection and all the lane marks are situated on the ground plane, we assume

that all image points are situated on the ground plane (this will have no influence on the lane mark features that we care about); with this assumption, we can compute the world counterpart of each image point—Via (3-2) with the constraint z_g=0, we can re-project the image onto the ground plane, in certain re-projection resolution as we need. In our implementation, each pixel in the bird-eye view image represents 0.1 meter; an example is shown in Fig.1.

If the pitch angle α can be estimated on-line, then it can be substituted into (3-2) to update $\mathbf{M}_\alpha$. If there is no such kind of on-line estimation mechanism, we can simply let α be 0 (considering that the pitch angle is generally small); the detection results are also acceptable for many applications.

3.2 Edge detection with low threshold

Edge points are extracted from the bird-eye view image using the Canny detector [17] in a modified way: first, image gradients are only computed along the vertical direction at each pixel because all lane marks are almost aligned in horizontal direction (this is a convenience brought by using the bird-eye view image); second, non-maximum suppression is carried out only along the vertical direction. This modified version is computationally more efficient than the original Canny detector.

As explained in previous section, we do not want to treat the edge detector threshold as one of the criteria to distinguish the lane marks apart from the background. We simply use a low threshold to guarantee the detection of 'any' lane mark edge points—it has to be admitted that it is difficult to guarantee the detection of absolutely 'any' lane mark edges, because a low threshold would still sift away some lane mark edge parts in adverse scenarios where these parts are even not so visible for our human eyes—Although a low threshold would incur comparatively more false positives, our philosophy is to reduce the false positives according to the criteria that are related to the inherent physical properties of the lane marks.

3.3 Edge point connection and edge segments matching

If any two edge points have the same gradient sign and they are in the neighbourhood of each other, then they are connected. In this way, a group of edge segments can be obtained. The segment length in the bird-eye view directly reflects the physical length of the segment (this is another convenience brought by the bird-eye view image). Since the length is an inherent physical property of the lane marks and it will not be influenced by the variation of road conditions. Therefore, we can fairly eliminate those edge segments whose length is below a certain value, as we know that there are construction standards which specify the minimum length of the lane marks. By so far, we intend to only use general *a priori* knowledge and do not want to get entangled with

any specific *a priori* knowledge associated with certain road region. Therefore, we set a conservative threshold of 1.0 meter, noting that many kinds of lane mark segments are much longer than this value. Obviously, if other sources of data besides the mono-vision can be used, such as a map or a GIS (Geographical Information System), then more pertinent length thresholds can be set for detecting the lane marks.

The next step after getting the edge segments is to match the edge segments to form lane mark segments (the final output of lane mark features). A lane mark segment always has the 'dark-bright-dark' pattern in its neighbourhood area; in other words, the up-side edge of a lane mark should always have the 'positive' sign while the down-side edge should always have the 'negative' sign. This pattern is another inherent physical property of the lane marks (noting that here we only care about the sign, not the value behind the sign).

Given an edge segment S_1 with positive sign, search in its down-side neighbourhood the closest edge segment with negative sign. The area of the neighbourhood under searching is determined by the width of the lane marks. If a qualified edge segment (denoted as S_2) can be found, then S_1 and S_2 are combined to form a lane mark segment which lies in the middle of these two segments; otherwise discard S_1. The lane width threshold is also set conservatively and set to be 0.8 meter, noting that the width of a lane mark is usually much smaller than this value.

By so far, the lane mark features used in our lane detection method, i.e. the lane mark segments, are finally obtained.

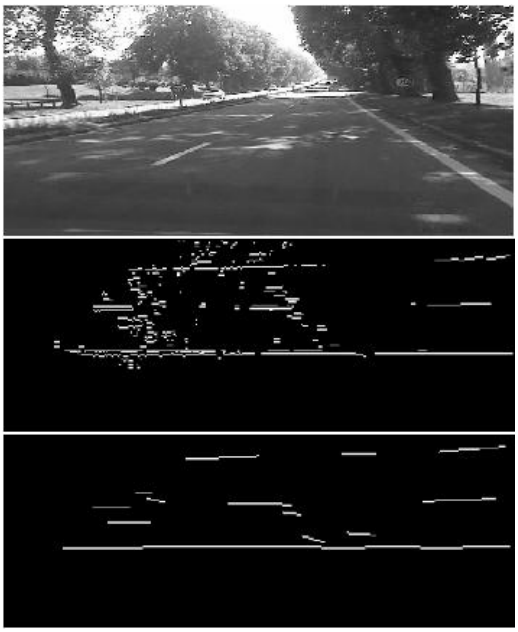

Figure 4 (top) the camera image; (middle) result of edge detection; (bottom) result of lane mark segment detection;

4 Lane Model Fitting

The task of lane model fitting is to propose possible lane configurations based on the lane mark segments extracted via the method described in previous section. The basic idea is: randomly generate a group of lane configuration samples and then evaluate the fitness of each sample.

4.1 Establish relationship among lane mark segments

As traversing all the possibilities in the model parameter space might be computationally forbidding, different methods in the spirit of random sampling such as the RANSAC method [12] have been proposed to search for semi-optimal solutions while keeping the computational expense at affordable level. To further save the computational expense, it would be better NOT to waste sampling on those absolutely impossible lane configurations. In other words, for any pair of lane mark segments, a preliminary judgement on whether they are possible to be on the same lane would contribute to computational efficiency.

Given two lane mark segments: S_1[A,B] and S_2[C,D]; where A and B are respectively the left and the right end points of S_1, C and D are respectively the left and the right end points of S_2. Furthermore, given that S_1 is on left to S_2, i.e. $B_x<C_x$. Suppose that these two lane mark segments are situated on the same lane and let the lane part between B and C be modelled by a cubic spline, whose direction at B is set the same to the direction from A to B and whose direction at C is set the same to the direction from C to D.

$$y = a_0 + a_1(x - B_x) + a_2(x - B_x)^2 + a_3(x - B_x)^3 \tag{4-1}$$

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{3}{(C_x - B_x)^2} & -\frac{2}{C_x - B_x} & \frac{3}{(C_x - B_x)^2} & -\frac{1}{C_x - B_x} \\ \frac{2}{(C_x - B_x)^3} & \frac{1}{(C_x - B_x)^2} & -\frac{2}{(C_x - B_x)^3} & \frac{1}{(C_x - B_x)^2} \end{bmatrix} \begin{bmatrix} B_y \\ \frac{B_y - A_y}{B_x - A_x} \\ C_y \\ \frac{D_y - C_y}{D_x - C_x} \end{bmatrix}$$

Compute the curvature at B, denoted as *curv*(B):

$$curv(B) = 2a_2 /(1+a_1^2)^{1.5} \approx 2a_2 /(1+1.5a_1^2)$$

By construction standards, the curvature of each road should not exceed the maximum curvature constraint. We set a conservative threshold for the curvature, which is set to be 0.04. If the curvature at B is no larger than this value, the lane mark segment S_2 is labelled to have 'right-side relationship' with the lane mark segment S_1; otherwise they do not have any relationship.

4.2 Lane configuration sampling

A group of lane configuration samples is generated randomly. Each sample is formed by selecting lane mark segments from the left-side (near-side to the vehicle) to the right-side in the following pseudo-code procedures:

-Randomly select a lane mark segment which is within a certain range ahead;
LOOP:
-If no lane mark segment has right-side relationship with the last selected lane mark segment, break the loop;
-Randomly generate a value from the set {T, F};
-If F, break the loop;
-If T, among those lane mark segments which have right-side relationship with the last selected lane mark segment, randomly select one;
LOOP END
LOOP: FOR any pair of neighbouring lane mark segments
-Fit a cubic spline to the two segments using (4-1);
-Give a flag to any lane mark segment which is between these two segments and is on the fitted cubic spline under a deviation tolerance;
LOOP END
-Add all lane mark segments with flag to current lane configuration sample;

Then a group of lane configuration samples is obtained, each of which consists of several lane mark segments or a single lane mark segment. Merge the lane configuration samples which are actually the same into one lane configuration sample, so as to remove the repetition of the samples. The fitness value of a lane configuration sample is the total length of all the lane mark segments in this sample.

5 Lane Tracking

The lane tracking process is carried out in the framework of the particle filter (or the sequential Monte Carlo method) [18], with a motivation to benefit from its potential to handle arbitrary distributions and multi-modal states.

5.1 Lane configuration model

Let the lane configuration be denoted as a generic function $f(x)$ expressed in the vehicle reference; at the beginning, we examine some mathematic properties of the $f(x)$.

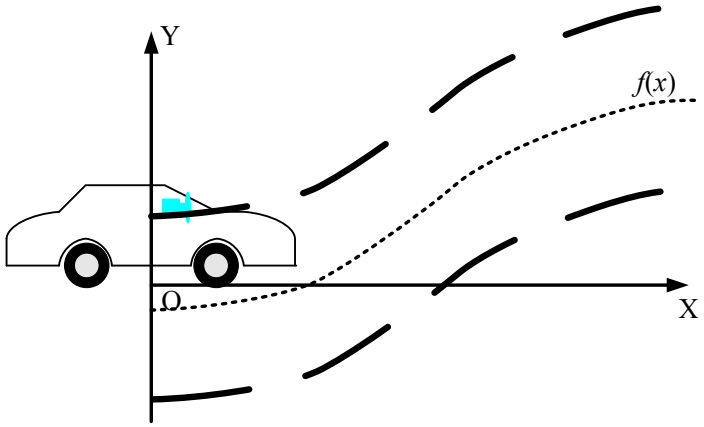

Figure 5 the lane configuration in the current vehicle reference

For the $f(x)$, suppose $f'^2 << 1$; in other words, suppose that the lane-vehicle relative orientation within a certain range in front of the vehicle is moderate. For example, even for a turn of the curvature radius of 120 meters, the relative orientation at a distance as far as 40 meters ahead is about only 1/3, whose square is about only 0.1. The curve at a point $(x, f(x))$, denoted as $curv(x, f(x))$ is:

$$curv(x, f(x)) = \frac{f''}{(1+f'^2)^{1.5}} \approx f'' \tag{5-1}$$

In other words, we can approximately regard the second-order derivative as the lane curvature. We can also approximate the curvature changing rate as follows:

$$\frac{d(curv(x, f(x)))}{dS} \approx \frac{df''}{dx} \approx f^{(3)} \tag{5-2}$$

Besides, we suppose that the curvature changing rate is constant—some construction standards also specify the maximum curvature changing rate; in other words, the lane is not allowed to jump from one curvature to another. This is consistent with the driving experience that we can not suddenly turn the driving wheel to a point but gradually make it—from (5-2) the third-order derivative of $f(x)$ can also be regarded as constant. Therefore, $f(x)$ can be fairly approximated as a three-order polynomial according to the Taylor expansion:

$$f(x) = f(0) + f'(0)x + \frac{f''(0)}{2}x^2 + \frac{f^{(3)}}{6}x^3 \qquad (5\text{-}3)$$

5.2 Lane state and state evolution model

5.2.1 Lane state

According to (5-3), the lane (within a certain range in front of the vehicle) can be modeled by a three-order polynomial, which is represented by four control points in our method (noting that four control points can uniquely determine a three-order polynomial). Let the four control points be denoted as (x_0,y_0), (x_1,y_1), (x_2,y_2) and (x_3,y_3); x_0 is set to be near the vehicle and x_3 is set to the maximum detection range; x_1 and x_2 are uniformly distributed between x_0 and x_3. The four control points are used to represent the center of the lane.

Besides the lane configuration, the lane width is also a factor to-be-considered. In reality, there are certain construction standards for the lanes and there is little variability for the lane width even in a rather large region. On the other hand, we do not intend to incorporate by so far any specific *a priori* knowledge that can be extracted from a pre-stored map. Therefore, we regard the lane width as a state to-be-estimated.

We could fairly use one state variable to denote current lane width, because of the parallelism of the lane boundaries; In contrast, we use two state variables to denote the lane width, namely w_0 and w_1, one of which represents the lane width at the near-vehicle limit of the lane detection range and the other represents that at the far-side limit; the lane width between the two limits is linearly interpolated by the two widths. In mathematical terms, let w_0 and w_1 respectively denote the lane width at (x_0,y_0) and (x_3,y_3); then the lane width at an arbitrary point (x,y) between (x_0,y_0) and (x_3,y_3) is given as:

$$w(x) = w_0 + \frac{w_1 - w_0}{x_3 - x_0}(x - x_0) \tag{5-4}$$

The motivation of using a redundant width is to account the influence of the vehicle pitch movement. More specifically, the parallelism of the lane boundaries might be distorted in the bird-eye view image, mainly due to the vehicle pitch movement. The use of the two widths can well characterize this distortion, as a benefit of which an on-line estimation of the vehicle pitch angle can be performed (This will be introduced latter).

In summary, a lane is completely modeled by four control points and two widths, i.e. $\{(x_0,y_0), (x_1,y_1), (x_2,y_2), (x_3,y_3), w_0, w_1\}$ and the task of lane tracking is to estimate them. Since x_0, x_1, x_2, and x_3 are always fixed, the actual task is to estimate the lane state $\mathbf{X}=(y_0, y_1, y_2, y_3, w_0, w_1)^T$.

5.2.2 State evolution model

At period k, let the lane configuration be denoted as $f_k(x)$ with the sub-script k. Le the vehicle displacement and yaw change be denoted respectively as Δs_k and $\Delta\theta_k$. We could assume that Δs_k and $\Delta\theta_k$ are close to zero; for example, for a period of 0.04 second, a vehicle velocity of 20 m/sec and a yaw rate of 0.2 (moving by 20 m/sec on a turn of the curvature radius of 100 meters), Δs_k and $\Delta\theta_k$ are respectively 0.8 meter and 0.008 rad. The relation between $f_{k-1}(x)$ and $f_k(x)$ is given as:

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} \approx \begin{bmatrix} \cos\Delta\theta_k & \sin\Delta\theta_k \\ -\sin\Delta\theta_k & \cos\Delta\theta_k \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \end{bmatrix} + \begin{bmatrix} -\Delta s_k \cos(\Delta\theta_k / 2) \\ \Delta s_k \sin(\Delta\theta_k / 2) \end{bmatrix} \tag{5-5}$$

Considering that $||y||$ is usually much smaller than $||x||$, (5-5) is approximated by a non-linear relation as in (5-6):

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} \approx \begin{bmatrix} x_{k-1} - \Delta s_k \\ y_{k-1} - \Delta\theta_k x_{k-1} \end{bmatrix} \tag{5-6}$$

Let $g(x)=f_k(x)-f_{k-1}(x)$, which reflects the lane configuration evolution from the perspective of the vehicle. From (5-6) we have:

$$g(x) \approx f_{k-1}(x + \Delta s_k) - \Delta\theta_k x - f_{k-1}(x) \approx f_{k-1}'(x)\Delta s_k - \Delta\theta_k x \tag{5-7}$$

Besides, we have:

$$
\begin{aligned}
&g(x_1) = g(x_0) + g'(\varepsilon_1)(x_1 - x_0) \\
&g(x_2) = g(x_0) + g'(\varepsilon_1)(x_2 - x_0) + (g'(\varepsilon_2) - g'(\varepsilon_1))(x_2 - x_1) \\
&= g(x_0) + g'(\varepsilon_1)(x_2 - x_0) + g''(\eta_1)(\varepsilon_2 - \varepsilon_1)(x_2 - x_1) \\
&g(x_3) = \\
&g(x_0) + g'(\varepsilon_1)(x_3 - x_0) + (g'(\varepsilon_2) - g'(\varepsilon_1))(x_3 - x_1) + (g'(\varepsilon_3) - g'(\varepsilon_2))(x_3 - x_2) \\
&= g(x_0) + g'(\varepsilon_1)(x_3 - x_0) + g''(\eta_1)(\varepsilon_2 - \varepsilon_1)(x_3 - x_1) + g''(\eta_2)(\varepsilon_3 - \varepsilon_2)(x_3 - x_2)
\end{aligned}
$$

The ε_1, ε_2, and ε_3 are certain values respectively between x_0 and x_1, between x_1 and x_2, and between x_2 and x_3; the η_1 and η_2 are certain values respectively between ε_1 and ε_2, and between ε_2 and ε_3. Then the evolution of the four control points is given as:

$$
\begin{aligned}
&y_{0,k} = y_{0,k-1} + \Delta y_{0,k} \\
&y_{1,k} = y_{1,k-1} + \Delta y_{0,k} + e_{1,k}(x_1 - x_0) \\
&y_{2,k} = y_{2,k-1} + \Delta y_{0,k} + e_{1,k}(x_2 - x_0) + e_{2,k}(x_2 - x_1) \\
&y_{3,k} = y_{3,k-1} + \Delta y_{0,k} + e_{1,k}(x_3 - x_0) + e_{2,k}(x_3 - x_1) + e_{3,k}(x_3 - x_2) \\
&\Delta y_{0,k} = g(x_0) \\
&e_{1,k} = g'(\varepsilon_1) \\
&e_{2,k} = g''(\eta_1)(\varepsilon_2 - \varepsilon_1) \\
&e_{3,k} = g''(\eta_2)(\varepsilon_3 - \varepsilon_2)
\end{aligned}
\qquad (5\text{-}8)
$$

As a result, the generative model of evolving $\{y_{0,k\text{-}1}, y_{1,k\text{-}1}, y_{2,k\text{-}1}, y_{3,k\text{-}1}\}$ to $\{y_{0,k}, y_{1,k}, y_{2,k}, y_{3,k}\}$ is given as:

$$
\begin{aligned}
&p(y_{0,k}, y_{1,k}, y_{2,k}, y_{3,k} \mid y_{0,k-1}, y_{1,k-1}, y_{2,k-1}, y_{3,k-1}) \\
&= p(y_{0,k} \mid y_{0,k-1}) p(y_{1,k} \mid y_{0,k-1}, y_{1,k-1}, y_{0,k}) \\
&\quad p(y_{2,k} \mid y_{0,k-1}, y_{1,k-1}, y_{2,k-1}, y_{0,k}, y_{1,k}) \\
&\quad p(y_{3,k} \mid y_{0,k-1}, y_{1,k-1}, y_{2,k-1}, y_{3,k-1}, y_{0,k}, y_{1,k}, y_{2,k}) \\
&= p(\Delta y_{0,k}) p(e_{1,k}) p(e_{2,k}) p(e_{3,k})
\end{aligned}
\qquad (5\text{-}9)
$$

From (5-7) we have:

$$\Delta y_{0,k} = g(x_0) \approx f_{k-1}'(x_0)\Delta s_k - \Delta\theta_k x_0$$
$$e_{1,k} = g'(\varepsilon_1) \approx f_{k-1}''(\varepsilon_1)\Delta s_k - \Delta\theta_k$$
$$e_{2,k} = g''(\eta_1)(\varepsilon_2 - \varepsilon_1) \approx f_{k-1}^{(3)}(\eta_1)\Delta s_k(\varepsilon_2 - \varepsilon_1)$$
$$e_{3,k} = g''(\eta_2)(\varepsilon_3 - \varepsilon_2) \approx f_{k-1}^{(3)}(\eta_2)\Delta s_k(\varepsilon_3 - \varepsilon_2)$$

The $\Delta y_{0,k}$, $e_{1,k}$, $e_{2,k}$, and $e_{3,k}$ can be treated as random variables, considering the range constraint of $f_{k\text{-}1}^{(2)}$ and $f_{k\text{-}1}^{(3)}$. The Δs_k and $\Delta\theta_k$ are also treated as random variables. As mentioned previously, in the works presented in this report, we only want to use mono-vision even without the aide of vehicle motion sensors. Therefore, Δs_k and $\Delta\theta_k$ are simply treated as zero-mean random variables with comparatively larger variance—we even do not want to take it for granted either that the vehicle is moving forward, though this is the usual case when lane detection is performed. In one word, by so far, we want to use as less specific *a priori* knowledge as possible. Of course, the availability of the motion data can better constrain the variance of Δs_k and $\Delta\theta_k$ (as will be concerned in the next report). The widths are simply varied randomly and the complete lane state evolution model can be written as:

$$\begin{aligned}
y_{0,k} &= y_{0,k-1} + \Delta y_{0,k} \\
y_{1,k} &= y_{1,k-1} + \Delta y_{0,k} + e_{1,k}(x_1 - x_0) \\
y_{2,k} &= y_{2,k-1} + \Delta y_{0,k} + e_{1,k}(x_2 - x_0) + e_{2,k}(x_2 - x_1) \\
y_{3,k} &= y_{3,k-1} + \Delta y_{0,k} + e_{1,k}(x_3 - x_0) + e_{2,k}(x_3 - x_1) + e_{3,k}(x_3 - x_2) \\
w_{0,k} &= w_{0,k-1} + \Delta w_{0,k} \\
w_{1,k} &= w_{1,k-1} + \Delta w_{1,k}
\end{aligned} \tag{5-10}$$

Then based on (5-9), the complete proposal distribution is modeled as:

$$\begin{aligned}
& p(\mathbf{X}_k \mid \mathbf{X}_{k-1}) \\
&= p(y_{0,k}, y_{1,k}, y_{2,k}, y_{3,k}, w_{0,k}, w_{1,k} \mid y_{0,k-1}, y_{1,k-1}, y_{2,k-1}, y_{3,k-1}, w_{0,k-1}, w_{1,k-1}) \\
&= p(y_{0,k}, y_{1,k}, y_{2,k}, y_{3,k} \mid y_{0,k-1}, y_{1,k-1}, y_{2,k-1}, y_{3,k-1}) p(w_{0,k} \mid w_{0,k-1}) p(w_{1,k} \mid w_{1,k-1}) \\
&= p(\Delta y_{0,k}) p(e_{1,k}) p(e_{2,k}) p(e_{3,k}) p(\Delta w_{0,k}) p(\Delta w_{1,k})
\end{aligned} \tag{5-11}$$

Noting that the vehicle pitch movement has little influence on the near-vehicle part of the bird-eye view image while having larger influence on the far-away part of the

image, the variance of $\Delta w_{0,k}$ is set comparatively small whereas that of $\Delta w_{1,k}$ is set comparatively large.

The lane state estimate consists of a set of *particles* $\{\mathbf{X}^i=(y_0^i, y_1^i, y_2^i, y_3^i, w_0^i, w_1^i)^T \mid i=1,2,\ldots,n\}$, each of which represents a lane configuration. Each particle is associated with a weight that specifies the probability of the particle being the true lane; let the weights be denoted as $\{q^i \mid i=1,2,\ldots,n\}$ The set of the particles at current period are generated from the set at the previous period, using the lane state evolution model (5-10). The model inputs $\Delta y_{0,k}$, $e_{1,k}$, $e_{2,k}$, $e_{3,k}$, $\Delta w_{0,k}$, and $\Delta w_{1,k}$ are randomly generated conforming to the proposal distribution model (5-11).

5.3 Lane state update

The lane configuration samples obtained by the method introduced in section 4 are used to update the lane state particles. Extract those lane configuration samples that are consistent with the lane particles under a specified distance tolerance; denote these extracted lane configuration samples as $\{\mathbf{S}^1, \mathbf{S}^2, \ldots, \mathbf{S}^m\}$, which are used to update the particles $\{\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^n\}$.

$$q_k^{\,i} = q_{k-1}^{\,i} \sum_{j=1}^{m} \exp(-\| \mathbf{X}_k^{\,i} - \mathbf{S}_k^{\,j} \|^2 / \sigma^2) \qquad (5\text{-}12)$$

The $\|\mathbf{X}_k^i - \mathbf{S}_k^j\|$ denotes a norm that represents the difference between the lane state particle $\mathbf{X}_k^i$ and the lane configuration sample $\mathbf{S}_k^j$. More specifically, for a lane configuration sample, we uniformly select several points on it and treat the summed squares of the distances of these points to the lane state particle as the norm. The weights of all the particles, after update, are normalized. If the distribution of the weights is not *uniform* enough (for example, if the inverse of the summed squares of the weights is below certain threshold), then one more step of *re-sampling* will take place, which draw new particles from current particles with their probability to-be-selected in proportion to their weights; the weights of all the new particles are reset to $1/n$. The *re-sampling* can also be performed every time after the *update*, as the case in our implementation.

5.4 Vehicle pitch angle estimation

The parallelism of the lane boundaries might be distorted in the bird-eye view image, mainly due to the vehicle pitch movement, as shown in Fig.6. The difference between the two widths w_0 and w_1 can well characterize this distortion, without which w_0 and w_1 should always be the same. By a geometric analysis, we can approximate recover the vehicle pitch angle α as follows:

$$\alpha_k = -\frac{w_{1,k} - w_{0,k}}{w_{0,k}} \cdot \frac{h}{L} \qquad (5\text{-}13)$$

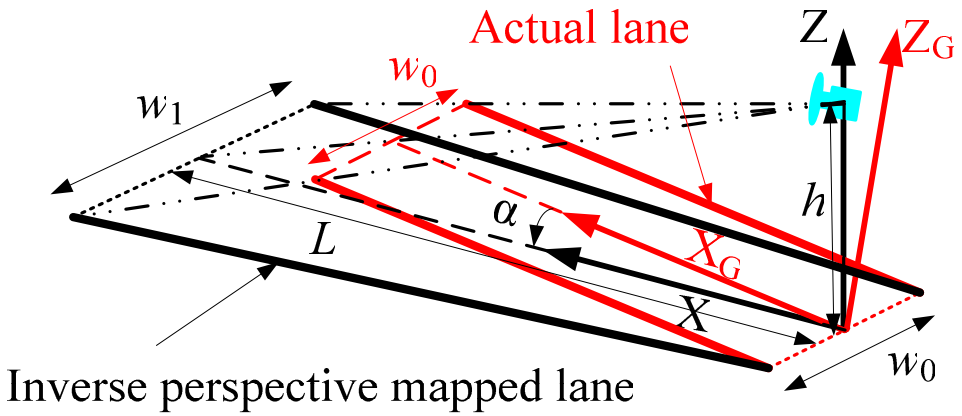

Figure 6 the distortion of the lane parallelism due to vehicle pitch angle

6 Experimentation

The tests were carried out based on a Citroen vehicle platform which is equipped with a mono-camera, two laser scanners, and motion sensors. For the moment, we only rely on the mono-vision to perform lane detection.

The function has been programmed in C++ in the *RTMaps* software platform [19]; the computation time for each image frame is less than 40 ms on an Intel Core2 2.0-GHz processor with a 2.0-GB RAM, noting that no coding optimization has been done yet and there is still room for enhancing computational efficiency.

First, the performance of the lane mark segment extraction method introduced in section 3 is demonstrated, in comparison with that of edge-detector based method. Fig.4 (top) shows a commonly encountered scenario where there are shadows caused by road-side trees. The Canny edge-detector is tried on the bird-eye view image; the threshold in the Canny edge-detector is tuned from low value to high value and a threshold value is chosen to have a suitable balance between the false-positives and true-negatives. Fig.4 (middle) demonstrates the result of edge detection associated with this threshold value. As shown in Fig.4 (middle), many lane mark edge parts have not been extracted while there are still a lot of noise edge points.

The introduced lane mark segment extraction method is also tried on the same bird-eye view image without any tuning of the parameters (as explained in section 3, a fixed low

threshold is always used); the result is shown in Fig.4 (bottom). As the result shows, the dashed lane mark in the middle of the road and the continuous lane mark on the right-side are completely detected without any miss-detection of their part. For the left-side road boundary which is almost invisible, despite some miss-detection, our method still extracts more portion of the lane boundary than what the edge-detector does. As the result also shows, the false-positives of the lane mark segment extraction method are much less than those incurred by the edge-detection method.

Next, the performance of the entire lane detection and tracking method is demonstrated. Thousands of structured road images have been logged for off-line testing, some example results of which are demonstrated in Fig.7.

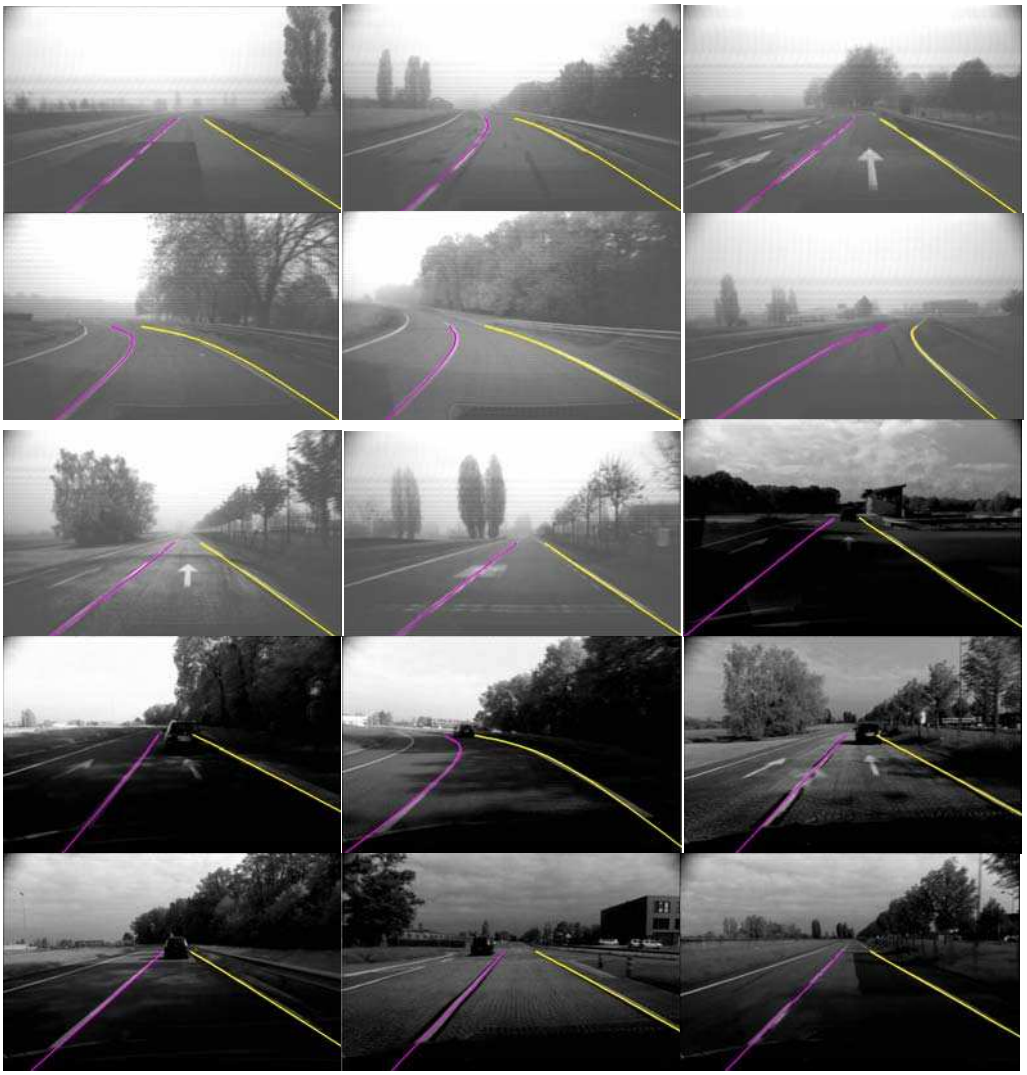

Figure 7 lane detection results

The road images under test include scenarios with large variability. For example, the lane configurations are diverse; there are straight roads as well as curved roads; there is variation in the width and length of the lane marks. Besides, the light condition changes; shadows caused by road-side trees and other road surface signs also appear from time to time. As can be see, the introduced method outputs desirable lane detection results for all these scenarios.

The effect of correcting the bird-eye view image based on the estimated vehicle pitch angles is demonstrated. Due to the vehicle pitch movement, the parallelism of the lane boundaries is distorted, as shown in the two bird-eye view images in the left column in Fig.8. On the other hand, the vehicle pitch angle is estimated using (5-13) and then used to correct the perspective matrix using (3-2). The same bird-eye view images are re-generated using the corrected perspective matrices; the effect of correction is shown in the two images in the right column in Fig.8. As we can see, the lane parallelism has been well recovered.

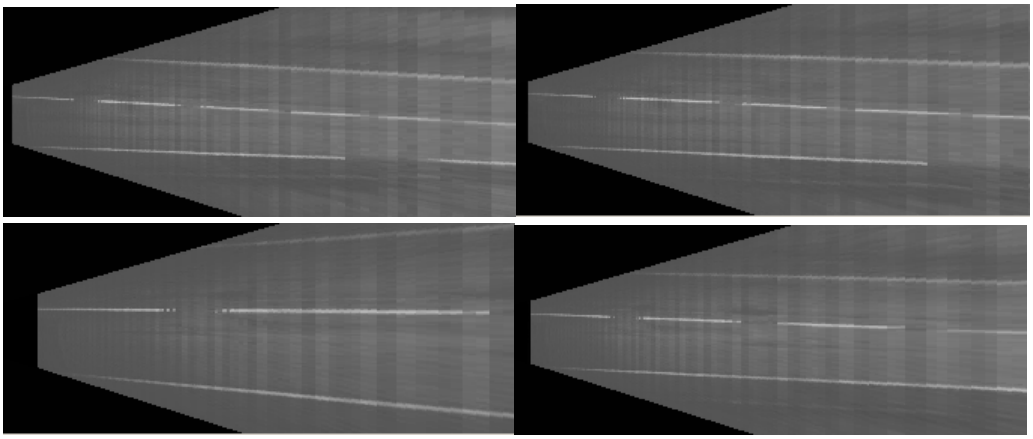

Figure 8 the bird-eye view image: (left) without correction; (right) with correction

7 Conclusion

We have introduced a mono-vision based lane detection method—strictly mono-vision without using any data of other sensors. The lane detection method consists of the sub-parts of lane mark segment extraction, lane model fitting, and lane tracking, the techniques of which are detailed. In lane mark segment extraction and lane model fitting, an essential spirit is to use some criteria that are only related to the inherent physical properties of the lane marks and to avoid employing specific a priori knowledge. The lane detection is performed in and the results are also expressed in the

world (the ground plane) reference instead of in the image reference; as a prerequisite of this practice, the techniques of inverse perspective mapping is detailed. The lane tracking is carried out in the framework of the particle filter; the lane configuration model and the lane state are presented, based on a geometric analysis. The techniques of the lane state evolution and the lane state update are also described. Besides, a method of vehicle pitch angle estimation is also given. The results of real-data tests are demonstrated, which shows promising performance of the introduced mono-vision based lane detection method.

On the other hand, the introduced lane detection method still allows much space for further improvements. As explained in section 2, the presented mono-vision based method is not intended to serve as a final solution but serve naturally as a base stone for extending the works into a multi-sensor fusion based methodology. Mono-vision has been reported to possess the ability to enhance the performance of other sensors [20]. Apparent improvements can also be envisioned for lane detection if a multi-sensor fusion based methodology is put into practice.

For example, we know that the lane marks are definitely situated on the ground plane and they are invisible for a laser scanner, then a laser scanner enables eliminating those image parts which can be associated to a range measurement. This can well exclude those challenging factors such as leading vehicles and road-side bars, whose parts sometimes appear similar and even the same to lane marks.

Besides, a GPS together with a map (even a rough map) can also contribute considerable to lane detection. They can constrain the range of the data to be processed. Moreover, they enable adaptive modeling; if we know that the vehicle is on a straight road, we can fairly reduce the lane configuration model from a three-order polynomial to a first-order line, which will not only save computational cost but also enhance robustness.

As our plan, the works of a multi-sensor fusion based methodology which exploits the mutual beneficial relationship between the mono-vision and some other sensors are going to be carried out and to be reported in a sister report "*Lane Detection (Part II): Multi-Sensor Fusion Based Method*".

Reference

[1] J.C. McCall, M.M. Trivedi, "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation", *IEEE Transactions on Intelligent Transportation Systems*, 7(1), 2006, pp.20-37

[2] Hao Li, "Cooperative perception: application in the context of outdoor intelligent vehicle systems", *Ph.D. Thesis, École Nationale Supérieure des Mines Paris (ParisTech)*, 2012

[3] D. Pomerleau, "Neural network vision for robot driving", *The Handbook of Brain Theory and Neural Networks, Cambridge, MIT Press*, 1995

[4] M. Bertozzi, A. Broggi, "GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection", *IEEE Transactions on Image Processing*, 7(1), 1998, pp.62-81

[5] R. Aufrere, "Reconnaissance et suivi de route par vision artificielle, application à l'aide à la conduite", Université de Clermont-Ferrand 2, 2001

[6] Z.W. Kim, "Robust lane detection and tracking in challenging scenarios", *IEEE Transactions on Intelligent Transportation Systems*, 9(1), 2008, pp.16-26

[7] Q. Li, N.N. Zheng, H. Cheng, "Springrobot: a prototype autonomous vehicle and its algorithms for lane detection", *IEEE Transactions on Intelligent Transportation Systems*, 5(4), 2004, pp.300-308

[8] M. Meuter, S. Muller-Schneiders, A. Mika, S. Hold, C. Nunn, A. Kummert, "A novel approach to lane detection and tracking", *IEEE Conference on Intelligent Transportation Systems*, 2009, pp.582-587

[9] K. Kluge, S. Lakshmanan, "A deformable template approach to lane detection", *IEEE Intelligent Vehicle*, 1995, pp.54-59

[10] Y. Wang, E.K. Teoh, D. Shen, "Lane detection and tracking using B-Snake", *Image and Vision Computing*, 22(4), 2004, pp.269-280

[11] M. Aly, "Real time detection of lane markers in urban streets", *IEEE Intelligent Vehicles Symposium*, 2008, pp.7-12

[12] M.A. Fischler, R.C. Bolles, "Random Sample Consensus: a paradigm for model fitting with applications to image analysis and automated cartography", *Communication of the ACM*, 24(6), 1981, pp.381-395

[13] R. Chapuis, R. Aufrere, F. Chausse, "Accurate road following and reconstruction by computer vision", *IEEE Transactions on Intelligent Transportation Systems*, 3(4), 2002, pp.261-270

[14] H. Li, F. Nashashibi, "Robust real-time lane detection based on lane mark segment features and general *a priori* knowledge", *IEEE International Conference on Robotics and Biomimetics*, 2011, pp.812-817

[15] Z. Zhang, "A flexible new technique for camera calibration", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11), 2000, pp.1330-1334

[16] O. Faugeras, “Three-dimensional computer vision: a geometric approach”, *Cambridge, MA:MIT Press*, 1993

[17] J. Canny, “A computational approach to edge detection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), 1986, pp.679-698

[18] A. Doucet, N. De Freitas, N. Gordon (eds.), “Sequential Monte Carlo methods in practice”, *New York: Springer-Verlag*, 2001

[19] F. Nashashibi, B. Steux, P. Coulombeau, C. Laurgeau, “RTMaps a framework for prototyping automotive multi-sensor applications”, *IEEE Intelligent Vehicles Symposium*, 2000, pp.99-103

[20] H. Li, F. Nashashibi, G. Toulminet, “Localization for intelligent vehicle by fusing mono-camera, low-cost GPS and map data”, *IEEE International Conference on Intelligent Transportation Systems*, 2010, pp.1657-1662

RESEARCH CENTRE
ROCQUENCOURT

Domaine de Voluceau
Rocquencourt BP 105
78153 Le Chesnay Cedex France

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr